Amazon EKS Workshop

(AWS Community Day’20 Pune)

Akash Agrawal | 07-02-2019

About Me

- Have 12+ years of experience in IT Industry
- Previously I worked with various clients like Sabre Travel Technologies / Citi Bank / Goldman Sachs / L&T Infotech etc.

- From last couple of years, I work as **Full Time Independent Consultant (Freelancer)**:
  - I work with different startups/enterprise helping them to design/improve their solutions around Kubernetes & Cloud (as Technical Adviser / Architect / Analyst)
  - Trainer around Cloud & Kubernetes Ecosystem

- Developer Community: https://www.meetup.com/Technology-Cafe-Bengaluru/

Containers

Containers

- **Package** your application & it’s dependencies (application dependencies)
- Container **Runtimes**: **Docker**, Rkt etc.

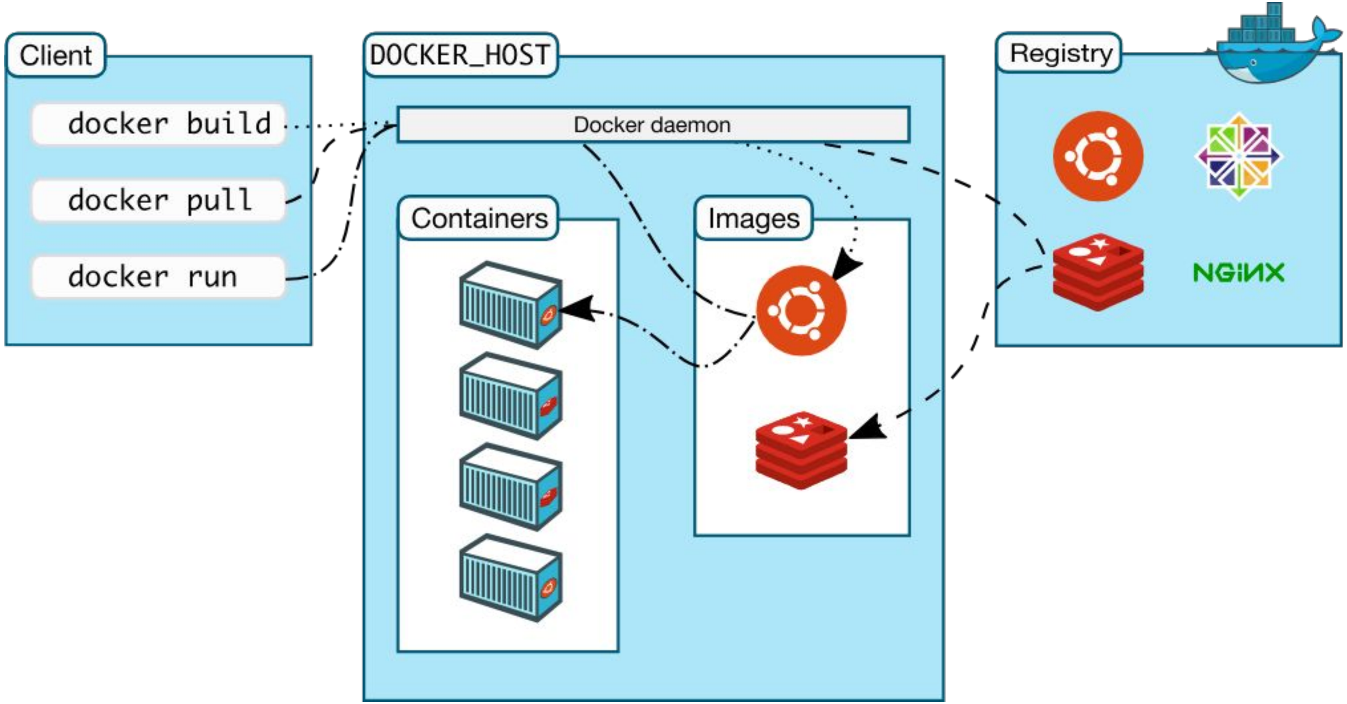
Docker

- It works on the principle of **Build / Ship / Run**
    - **Build:** Build Container Images on local Laptop or automate it through Continuous Integration Pipeline & Build Process **(Container Images)**
    - **Ship:** Ship Container Images through Container Registry e.g. DockerHub, ECR etc **(Container Registry)**
    - **Run:** Run anywhere on Cloud/On-Prem etc. **(Containers)**

Docker Terminology

- Docker Engine (Container Runtime)
- Docker Images (Container Snapshots) + Dockerfile
- Docker Registry (Container Storage)
- Containers (Runtime entity)

Docker


Client
docker build
docker pull
docker run
DOCKER_HOST
Docker daemon
Containers
Images
Registry
NGINX

Containers (Impact)

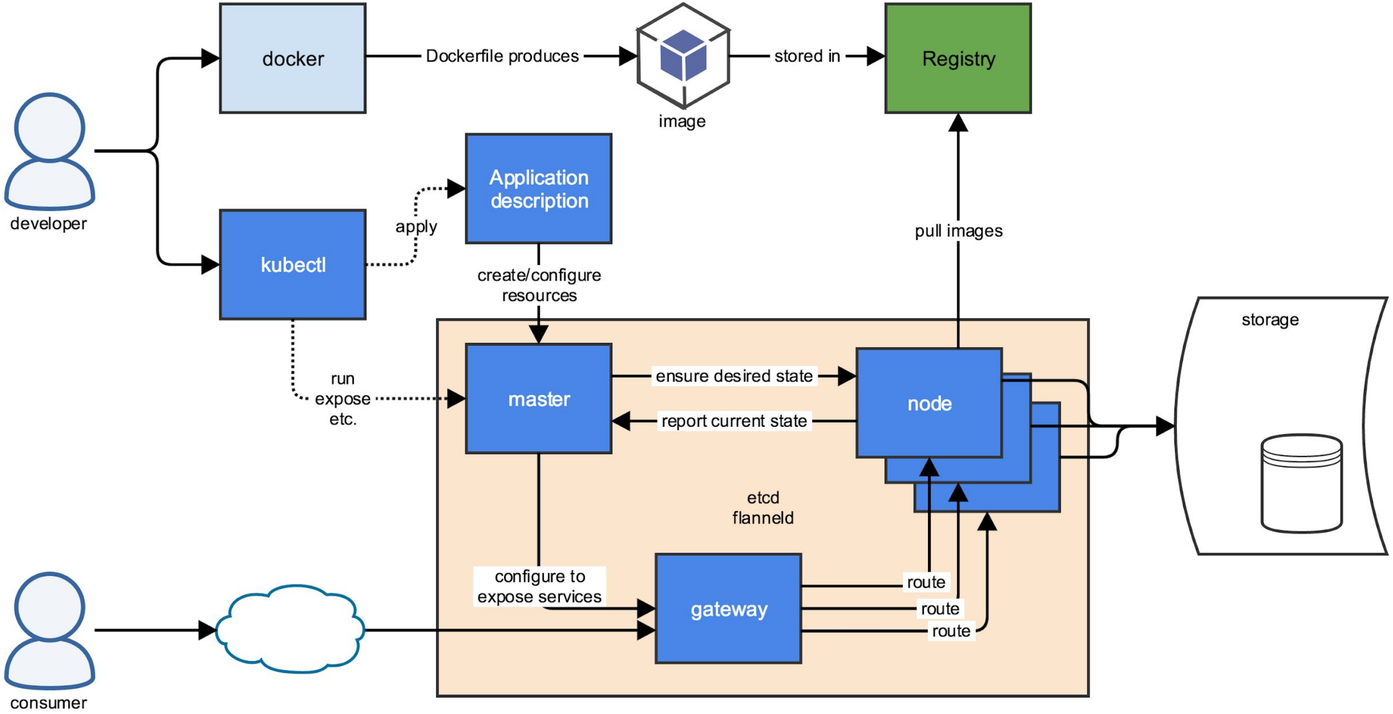
- Environment Consistency & Stability
- Machine (+ dependency) & Application (+ dependency) Separation - containers leave no/least footprints after removal
- Better Resource Utilization & Density - think from provider’s point of view
- Lightweight - distribution is easy, explore use cases like CI/CD pipelines & microservice architecture based applications

Kubernetes (Open Source)

Kubernetes

- Open Source System for Container Orchestration (Deployment/Scaling/Management)

- Focused around:
  - Scheduling workloads
  - Scaling (Manual / Auto) workloads
  - Self Healing workloads
  - ...



Source: https://software.danielwatrous.com/wp-content/uploads/2017/02/kubernetes-overview.png

Kubernetes Components

- Control Plane (Master Node) Components
  - API Server
  - Scheduler
  - Controller Manager
  - Etcd

- Data Plane (Worker Node) Components
  - Kubelet
  - Kube-Proxy
  - Container Runtime

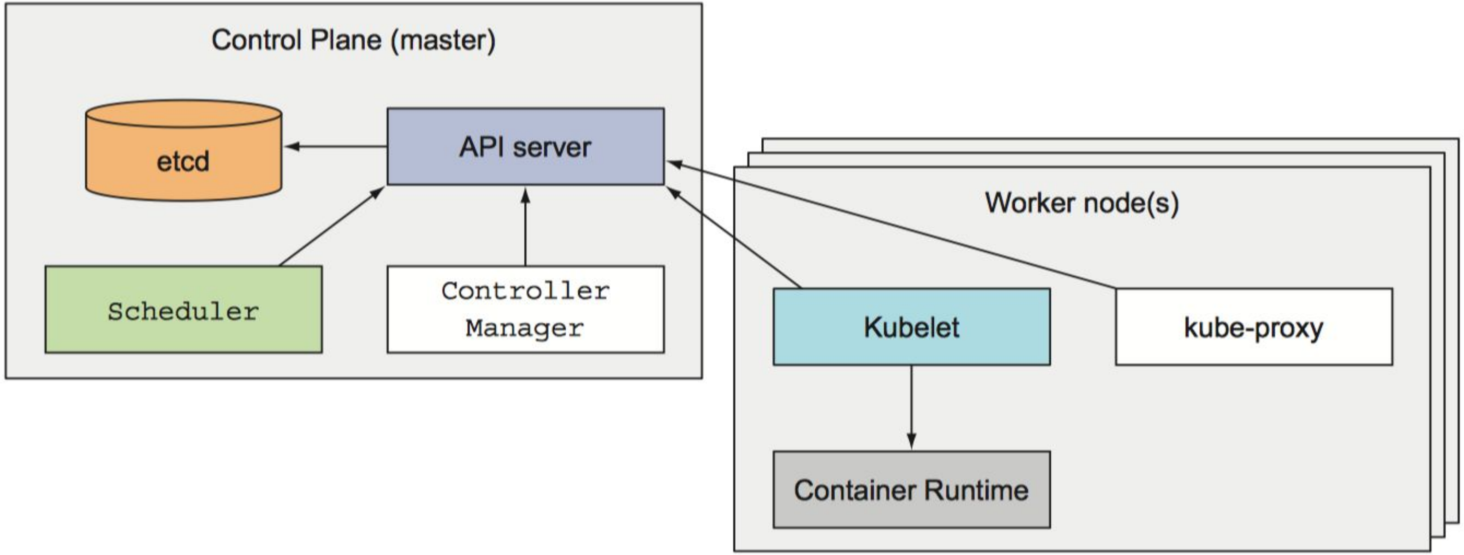


Figure 1.9 The components that make up a Kubernetes cluster

Source: https://carltsuis-blog.readthedocs.io/en/latest/kubernetes/components-of-k8s.png

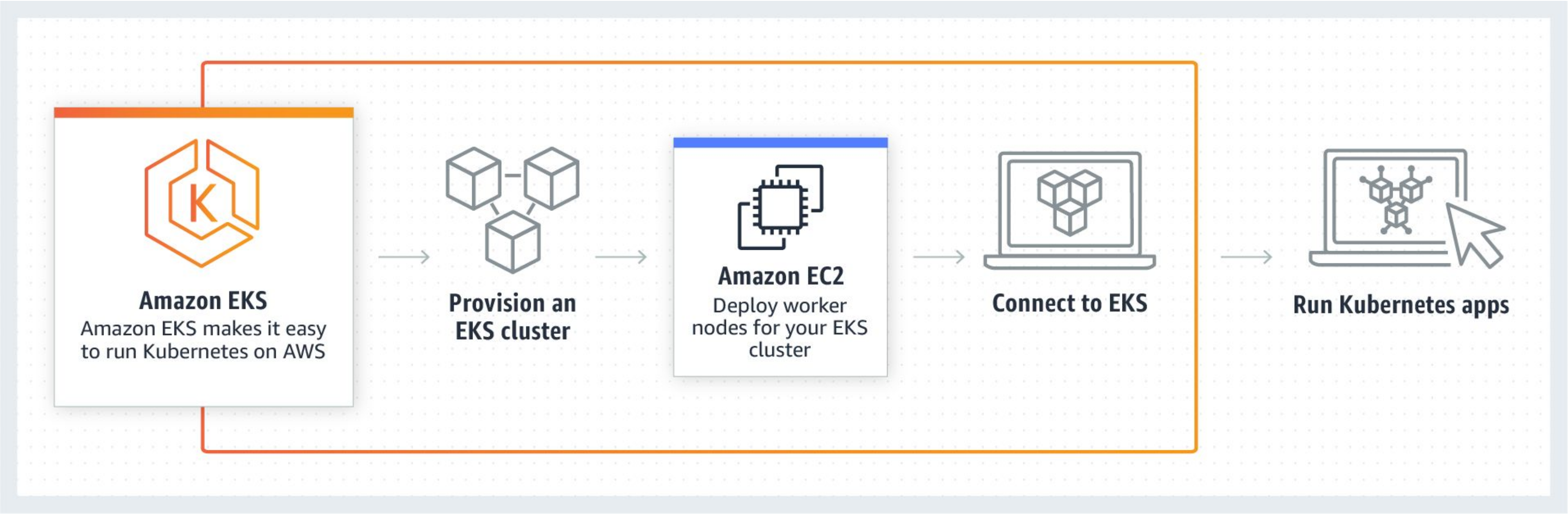
Amazon EKS

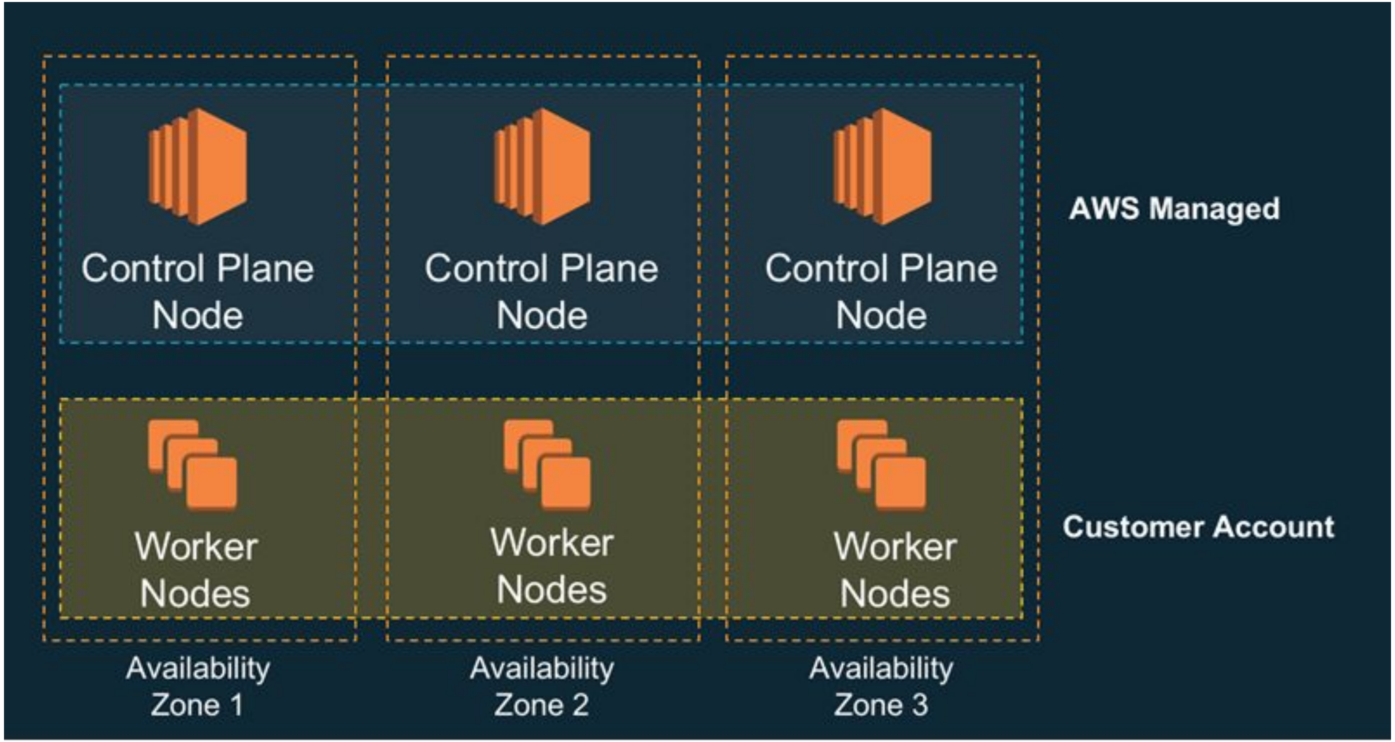
Kubernetes on AWS

- Bootstrap Kubernetes - The hard way
- Kubeadm
- Kops
- Amazon EKS

Amazon EKS

- Managed Kubernetes Service on AWS Cloud
- Customer should not worry about managing Control Plane, but of course they can access it
- Certified Kubernetes Conformant
- Fast & Reliable Kubernetes Cluster (Kubernetes System) - Setup & Maintenance (especially for Production Level workloads)

Amazon EKS
Amazon EKS makes it easy to run Kubernetes on AWS
Provision an EKS cluster
Amazon EC2
Deploy worker nodes for your EKS cluster
Connect to EKS
Run Kubernetes apps



Source: https://bekitzur.com/wp-content/uploads/2018/06/wsi-imageoptim-amazon-eks-2.png

Amazon EKS - Features

- Control Plane HA (in multiple AZs)
- Auto Repairing / Patching of Control Plane Nodes
- EKS Optimized AMIs (also with GPU Support)
- Integration:
  - Amazon ECR (for Container Registry)
  - Load Balancers (CLB/NLB/ALB)
  - Authentication & Authorization (through AWS IAM, Kubernetes RBAC)
  - Network Isolation (through VPC & Subnets)
  - ...

Amazon EKS - Day 1 Job

- Using CLI (on client/laptop)
  - Install AWS CLI (on client/laptop)
  - Connect it using AWS Access Key ID / Secret Access Key (on client/laptop)
  - Install eksctl
  - Install kubectl

- Follow the steps:
  - https://eksctl.io/usage/creating-and-managing-clusters/
    OR
  - https://docs.aws.amazon.com/eks/latest/userguide/getting-started-eksctl.html

Amazon EKS - Day 1 Job

- eksctl command will something look like:

  ```
  eksctl create cluster \
  --name prod \
  --version 1.13 \
  --nodegroup-name standard-workers \
  --node-type t3.medium \
  --nodes 3 \
  --nodes-min 1 \
  --nodes-max 4 \
  --node-ami auto
  ```

Amazon EKS - Day 1 Job

- Play around different eksctl capabilities:
    - https://eksctl.io/

Kubernetes for different workloads

Kubernetes Workloads

- Stateless Applications
- Stateful Applications
- Daemon Processes
- Batch Jobs

Kubernetes Resources/Objects

- Pods (Single & Multi Container Pods)
- Controllers:
  - Deployment / ReplicaSet / Replication Controller
  - StatefulSet
  - Job
  - DaemonSet
  - ...
- Services:
  - Cluster IP / NodePort / LoadBalancer

Kubernetes Resources/Objects

- Namespace
- ConfigMap
- Secret

Stateless Workloads

- Use Deployment Controller

Stateful Workloads

- Think about:
  - Data
  - Infrastructure & Operational Logic
- Use StatefulSet Controller

Data Persistence / State Management

- Containers are ephemeral in nature, so they lose data during container's destroy/create or restart cycles
- Any persistent data must be stored outside of container

Data Persistence Medium

- Pod Level
- Host Level
- Remote:
    - Cloud based Block storage solutions e.g. EBS volumes etc.
    - Network File Systems e.g. NFS etc.
    - Third Party Storage Systems e.g. GlusterFS etc.
    - Cloud Native Storage Solutions e.g. OpenEBS etc.

Data Persistence / State Management … cont.

- Docker achieves Data Persistence with Volume & Volume Plugins
- Kubernetes achieves or extends the same with:
    - Volumes (Kubernetes Volumes) & Volume Types/Plugins
    - Persistent Volume (PV)
    - Persistent Volume Claim (PVC)
    - Persistent Volume Type/Plugin
    - Storage Class

StatefulSets

- StatefulSet is intended to be used for Stateful applications
- HA & FT Stateful workloads mostly have certain requirements like:
    - Stable Unique Identifiers
    - Stable Persistent Storage
    - Ordered, graceful deployment & scaling
    - Ordered, graceful deletion & termination

StatefulSets ... cont.

- Stateful Pods have Unique identity. Identity sticks to the pod, regardless of which node it is scheduled on.
- PVs (associated with PVCs) are not get deleted, when you delete pods/statefulsets. It ensures Data Safety.
- StatefulSets require a Headless Service to be responsible for the network identity of the pods

StatefulSets … cont.

- Deployment & Scaling guarantees:
  - For a StatefulSet with N replicas, when pods are being deployed, they are being created sequentially (in order from 0 … N-1)
  - When Pods are being deleted, they are terminated in reverse order (in order from N-1 … 0)
  - Before a scaling operation is applied to a pod, all of its predecessors must be running & ready
  - Before a pod is terminated, all of its successors must be completely shut down.

Best Practices & Design

Overall Impact (Containers + K8s + Amazon EKS)

- Technical Impact:
  - Building, Distributing & Managing Containerized Applications at Scale
- Business Impact:
  - DevOps:
    - Infrastructure
    - CI/CD Pipelines
  - Product (Developers):
    - Distributed Systems (eg. MicroService Architecture based applications)
  - Providers (IaaS/PaaS/SaaS)
    - Better Infrastructure Utilization & Resource Isolation (for multiple tenants)

Best Practices

- Container Best Practices
- Kubernetes Best Practices
- GKE Best Practices

Architecture & Design

- Cluster Design
- Application/Solution Design

Amazon EKS - Day 2 Job

- Go to eksworkshop & look for your use case:
  - https://eksworkshop.com/

- Explore different Integrations e.g.:
  - MicroService Architecture
  - Service Mesh setup : with Istio/AWS App Mesh
  - DevOps:
    - IAC : CloudFormation / Terraform
    - CM : Ansible
    - CI/CD Pipeline : AWS Code Pipeline etc.

Amazon EKS - Day 2 Job

- Explore EKS SLA
  - https://aws.amazon.com/eks/sla/

- Explore EKS in scope by Compliance Program
  - https://aws.amazon.com/compliance/services-in-scope/

Thanks

Akash Agrawal

LinkedIn: http://www.linkedin.com/in/akash-agrawal-58a97813
Twitter: @akkiagrawal29